

# Strukture podataka i algoritmi

## C# osnove

dr. sc. Hrvoje Kalinić

# Ponavljjanje

- Klasa
- Konstruktor
  - preopterećeni
- Metoda
  - Varijabilni ulazi i izlazi
- Objekt
  - Instanciranje konstruktorom
- get i set
- Niz, lista
  - dohvat i postavljanje
- Prosljeđivanje parametara
  - "by reference"
  - "by value"
- Rad sa stringovima
  - Pretvaranje u znakove

# Ponavljjanje

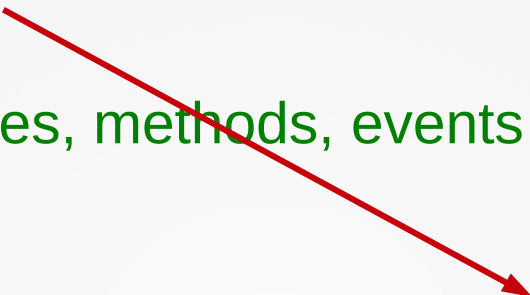
- Klasa
- Konstruktor
  - preopterećeni
- Metoda
  - Varijabilni ulazi i izlazi
- Objekt
  - Instanciranje konstruktorom
- get i set
- Niz, lista
  - dohvat i postavljanje
- Prosljeđivanje parametara
  - "by reference"
  - "by value"
- Rad sa stringovima
  - Pretvaranje u znakove

# Klasa

```
public class Lopta  
{  
    //Fields, properties, methods, events  
}
```

# Klasa

```
public class Lopta  
{  
    //Fields, properties, methods, events  
}
```



Kupac  
Zaposlenik  
Osoba  
Zgrada  
Stan  
Vozilo  
Automobil  
Zrakoplov  
...

# Klasa

```
public class Lopta  
{  
    //Fields, properties, methods, events  
}
```

Da, ovo je  
klasa...

Kupac  
Zaposlenik  
Osoba  
Zgrada  
Stan  
Vozilo  
Automobil  
Zrakoplov  
...

# Klasa

```
public class Lopta  
{  
    //Fields, properties, methods, events  
}
```

... i ona je javna.  
= svatko može napraviti  
objekt ove klase

Da, ovo je  
klasa...

Kupac  
Zaposlenik  
Osoba  
Zgrada  
Stan  
Vozilo  
Automobil  
Zrakoplov  
...

# Klasa

```
public class Lopta  
{  
    //Fields, properties, methods, events  
}
```

... i ona je javna.  
= svatko može napraviti  
objekt ove klase  
???

Da, ovo je  
klasa...

Kupac  
Zaposlenik  
Osoba  
Zgrada  
Stan  
Vozilo  
Automobil  
Zrakoplov  
...



# Klasa

```
public class Lopta  
{  
    //Fields, properties, methods, events  
}
```

"varijabla" definirana unutar klase  
(spremnik)

# Klasa

```
public class Lopta
{
    // private field
    private int tezina;

    // public field (ne preporuča se)
    public string materijal;
}
```

# Klasa

```
public class Lopta
{
    // private field
    private int tezina;

    // public field (ne preporuča se)
    public string materijal;
}
```

# Klasa

```
public class Lopta  
{  
    //Fields, properties, methods, events  
}
```

Posebna metoda pristupa  
(eng. accessor)  
- ponaša se kao varijabla

# Klasa

```
public class Lopta
{
    public int Tezina
    {
        get { return tezina; }
        set
        {
            // postavi neke smislene granice (npr u gramima)
            if (value > 0 && value <= 5000)
            {
                tezina = value;
            }
            else
                throw new ArgumentOutOfRangeException();
        }
    }
}
```

# Klasa

```
Lopta lopta = new lopta();  
lopta.Tezina = 985; // pozvan set pristup  
System.Console.Write(lopta.Tezina); // pozvan get pristup
```

# Klasa

```
public class Lopta  
{  
    //Fields, properties, methods, events  
}
```

a.k.a funkcija()



# Klasa

```
public class Lopta
{
    public void SetTezina(int value)
    {
        // postavi neke smislene granice (npr u gramima)
        if (value > 0 && value <= 5000)
        {
            tezina = value;
        }
        else
            throw new ArgumentOutOfRangeException();
    }
}
```

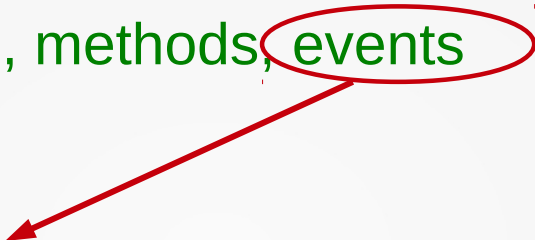


# Klasa

```
Lopta lopta = new lopta();  
lopta.SetTezina(985); // pozvana metoda SetTezina
```

# Klasa

```
public class Lopta  
{  
    //Fields, properties, methods, events  
}
```



Način da jedan objekt (ili klasa) obavijesti drugi objekt (ili klasu) o "zanimljivom događaju"

# Klasa

```
class Lopta_za_djecu : Lopta
{
    public bool za_djecu_je {get; set;}
}
```

# Klasa

```
class Lopta_za_djecu : Lopta
{
    public bool za_djecu_je {get; set;}
}
```

Nasljeđivanje klasa

# Ponavljjanje

- Klasa
- Konstruktor
  - preopterećeni
- Metoda
  - Varijabilni ulazi i izlazi
- Objekt
  - instanciranje
  - konstruktorom
- get i set
- Niz, lista
  - dohvat i postavljanje
- Prosljeđivanje parametara
  - "by reference"
  - "by value"
- Rad sa stringovima
  - Pretvaranje u znakove

# Konstruktor

```
public class Lopta
{
    public int tezina;
    public string materijal;
    public Lopta()
    {
        postoji = true;
        materijal = "guma";
        tezina = 200;
    }
}
```

# Konstruktor

```
public class Lopta
{
    public int tezina;
    public string materijal;
    public Lopta()
    {
        postoji = true;
        materijal = "guma";
        tezina = 200;
    }
}
```

# Konstruktor

```
class TestLopta
{
    static void Main()
    {
        Lopta L = new Lopta();
        Console.WriteLine(L.postoji);
    }
}
```



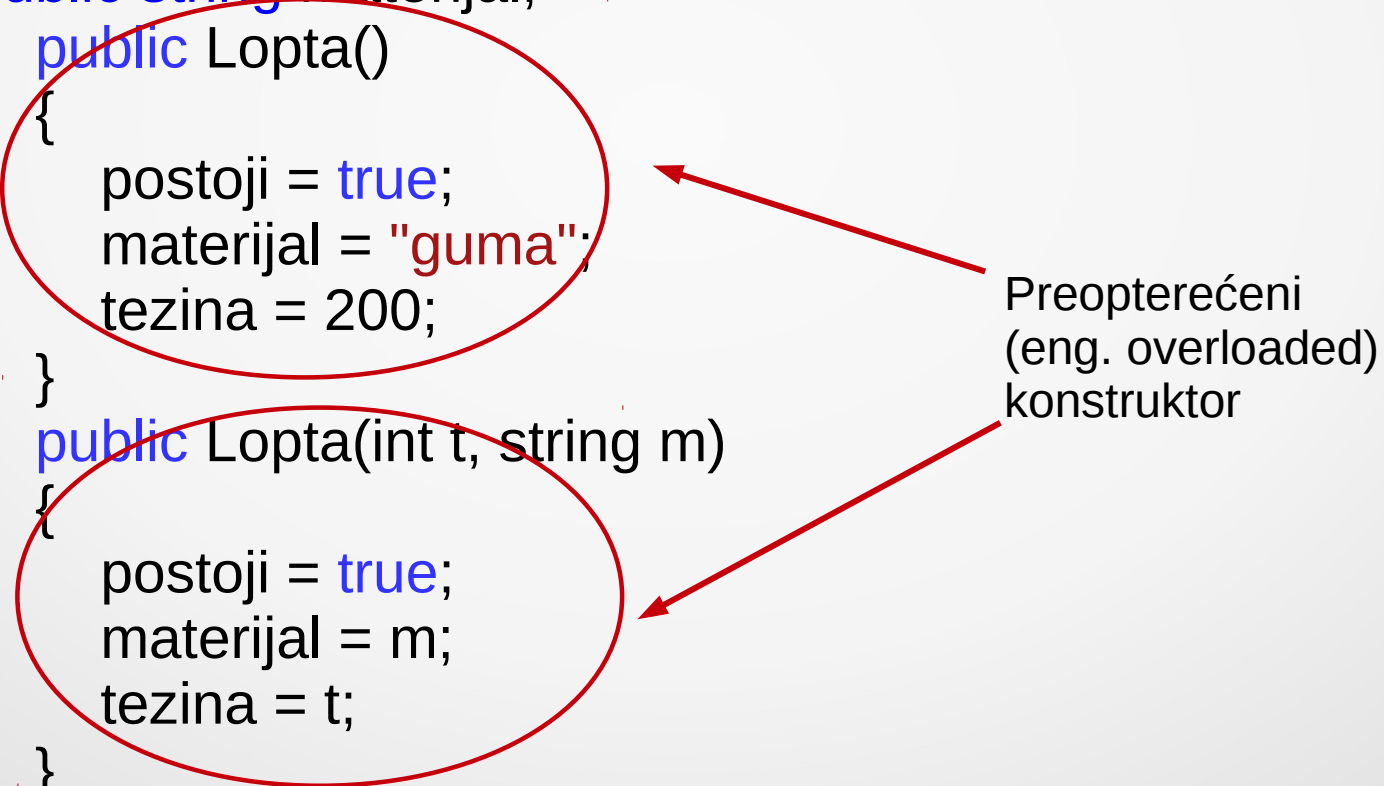
# Konstruktor

```
public class Lopta
{
    public int tezina;
    public string materijal;
    public Lopta()
    {
        postoji = true;
        materijal = "guma";
        tezina = 200;
    }
    public Lopta(int t, string m)
    {
        postoji = true;
        materijal = m;
        tezina = t;
    }
}
```

# Konstruktor

```
public class Lopta
{
    public int tezina;
    public string materijal;
    public Lopta()
    {
        postoji = true;
        materijal = "guma";
        tezina = 200;
    }
    public Lopta(int t, string m)
    {
        postoji = true;
        materijal = m;
        tezina = t;
    }
}
```

Preopterećeni  
(eng. overloaded)  
konstruktor



# Konstruktor

```
class TestLopta
{
    static void Main()
    {
        Lopta L1 = new Lopta();
        Lopta L2 = new Lopta(323,"plastika");
        Console.WriteLine("Tezina lopte L1: {0}", L1.tezina)
        Console.WriteLine("Tezina lopte L2: {0}", L2.tezina)
    }
}
```

